



Queries N + 1

Transcripción

[00:00] Hola, el cliente está contento con lo que estamos haciendo para mejorar nuestra aplicación de control de stock, ya tenemos las funcionalidades principales acá para registrar los productos y en la clase anterior nosotros realizamos el registro de un producto con categoría.

[00:16] Para finalizar nuestra aplicación falta la funcionalidad de reporte para listar a todos los productos agrupados por la categoría. Ahora cuando hacemos un clic acá en el botón ver reporte nos abre una pantalla vacía. Aquí la idea es listar de forma sencilla todos los productos con sus categorías.

[00:35] Vamos al código para ver en dónde tenemos que agregar esta lógica. Aquí en la clase ControlDeStockFrame nosotros tenemos aquí el botón, a ver, botón reporte. Vamos al método configurarAccionesDelFormulario para ver cómo está su configuración.

[00:51] Aquí en el botón reporte tenemos la acción acá y esta acción llama al método abrirReporte. Si entramos a este método nosotros vemos que estamos inicializando aquí un new ReporteFrame enviando como referencia el frame que lo está llamando. Vamos a entrar entonces aquí a ReporteFrame para ver qué hace.

[01:14] Aquí en la clase ReporteFrame, nosotros tenemos un constructor en donde estamos iniciando la pantalla y también tenemos un método cargarReporte en donde tenemos aquí un TODO que va a hacer en dónde vamos a estar agregando la lógica. Aquí también en este método ya tenemos

aquí un contenido, una variable de contenido que recibe como resultado acá el retorno del método cargarReporte de la categoría controller.

[01:43] Entonces vamos a entrar ahí a este método para ver qué tenemos que hacer. Bueno, aquí en este método está justamente comentado con el TODO y nosotros tenemos que justamente devolver nuestra lista de productos con categorías para el método de ReporteFrame. Aquí lo que vamos a hacer inicialmente va a ser así.

[02:04] Vamos a ver cómo funciona primero. Aquí vamos a estar devolviendo un `this.listar()`; para devolver primero todas las categorías. Entonces aquí cambiamos el tipo para categoría. Y por ahora, listo. Ahora en el reporteFrame nosotros vamos a borrar ese TODO, porque ya estamos empezando a hacer la lógica.

[02:26] Y aquí adentro del contenido, en cada fila que vamos a estar aquí creando nosotros vamos a agregar una propia fila aquí, justamente, vamos a cambiar este nombre para categoría, así queda más legible y entendemos que es que estamos agregando. Okay, bueno, voy a guardar estos cambios y voy a reiniciar la aplicación.

[02:49] La aplicación ya se reinició, ahora vamos a hacer un clic acá en ver reporte. Y ahora tenemos todas las categorías acá que tenemos registradas en la base, o sea ya funciona algo. Ahora tenemos que agregar a los productos. Para agregar a los productos, nosotros tenemos que buscar a los productos ¿con quién? Con el productoController.

[03:08] Entonces voy a crear un atributo acá, en el reporteFrame de productoController. Y aquí se llama productoController; voy a importar la clase, ahí está. Y en el constructor vamos a inicializarlo también: `this.productoController = new ProductoController()`; ahí está, pero hasta ahora, si venimos aquí a productoController nosotros tenemos como listar a todos los productos de la base.

[03:40] Ahora necesitamos tener una forma de buscar a los productos por categoría, porque aquí en el `reporteFrame`, nosotros hacemos un `forEach` de la categoría, entonces tenemos que para cada categoría buscar a sus productos para mostrar en el reporte. Como el producto tiene una columna que hace referencia a la categoría, nosotros podemos hacer esta búsqueda.

[04:03] Lo que tenemos que hacer entonces es aquí en `productoController` crear un método que haga una búsqueda de los productos por la categoría, entonces vamos a hacer lo siguiente. Hagamos aquí un `public` que devuelve un `List` de productos, entonces `producto`, aquí es el tipo, y va a ser también un nombre `listar`, pero como parámetro acabamos a sobrecargar el método.

[04:29] Como parámetro vamos a estar recibiendo una categoría porque queremos buscar al producto por la categoría. Entonces, aquí está la categoría. Muy bien. Y para devolver este resultado ¿qué vamos a hacer? Un `return productoDAO`, porque es quien hace quien realiza el acceso a la base de datos. `Listar`, enviando la `categoría.getId()`.

[04:58] Ahí enviamos el `id` de la categoría y queremos buscar a todos los productos por el `id` de la categoría. Entonces aquí vamos a crear el método. Ya lo creamos. Y este método va a tener prácticamente la misma lógica del método del listado. Entonces, para ahorrar un poco de tiempo, voy a justamente copiar toda esta lógica de acá y vamos a hacer un cambio bastante sencillo.

[05:27] Entonces aquí voy a sobrescribir esa parte, ahora sí. Acá voy a cambiar para `categoríaId` el nombre del parámetro y en donde tenemos el "`SELECT ID, NOMBRE, DESCRIPCIÓN, CANTIDAD FROM PRODUCTO`", vamos a hacer un "`FROM PRODUCTO`" `WHERE`, y acá vamos a agregar el `WHERE CATEGORÍA_ID = ?`")

[05:59] Lo que tenemos que hacer ahora es aquí, adentro de `try (statement)` hacer un `statement.setInt` en donde agregamos en el primer parámetro porque

tenemos una sola interrogación, la categoríaId. Así estamos haciendo la búsqueda de productos por la categoría. Todo el restante de la lógica queda igual.

[06:22] Entonces ya tenemos cómo hacer la búsqueda de los productos por el id de la categoría. Okay. Terminamos aquí con productoDAO y productoController. Ahora volvemos a reporteFrame aquí en el método cargarReporte, aquí lo podemos finalizar de la siguiente forma. Para cada categoría que tenemos nosotros vamos a hacer la búsqueda de los productos en el productoController para poder agregar acá en el listado también como una nueva fila.

[06:50] Entonces aquí voy a incrementar un poco más este forEach, déjame formatearlo, ahora sí, y tenemos aquí estamos agregando la categoría que va a ser nuestro título de la línea, digamos una fila título, y para cada producto vamos a hacer un var productos. Déjame bajar un poquito. Ahí.

[07:18] Va a recibir el resultado de this.productoController.listar enviando la categoría. Ahí está, ya tenemos el listado de productos. ¿Ahora qué tenemos que hacer? Otro forEach. Entonces productos.forEach(producto) ¿Y qué vamos a hacer con eso? Vamos a hacer igual que aquí, un modelo.

[07:48] Modelo.addRow. En el addRow vamos a estar agregando lo siguiente. Vamos a, ahora sí me ayuda el Eclipse, vamos a hacer un new Object, así como hicimos arriba. Y acá ponemos los siguientes valores: ponemos una columna vacía porque no quiero que en la primera columna del reporte aparezca como muy abajo, entonces una columna vacía.

[08:24] Después producto.getNombre() y producto.getCantidad(). Estos son los valores que vamos a estar utilizando acá. Entonces, tenemos tres valores, perdón, dos valores, una columna vacía y dos columnas: una con el nombre y la otra con la cantidad de productos. Ahí está.

[08:48] Terminamos la lógica para hacer nuestro reporte. Ahora vamos a reiniciar la aplicación para probar el código una vez más. Se levantó la aplicación, ahora hacemos un clic acá en ver reporte y acá tenemos el reporte, o sea, tenemos aquí por cada categoría: muebles, tecnología, cocina, zapatillas, tenemos a sus productos también.

[09:10] Muy bien, no es de los reportes más lindos, pero es funcional. Llegamos a un resultado, tenemos la funcionalidad de reporte, aquí, para cada categoría que tenemos en la base de datos estamos haciendo una búsqueda de la lista de productos para poder imprimir en la pantalla. Si venimos aquí en el `reporteFrame` podemos ver que en el cargar reporte y en el listar abrimos dos conexiones a la base de datos.

[09:36] Entonces con dos conexiones a la base de datos, nosotros logramos buscar todas las informaciones de nuestro stock de la tienda. ¿Pero será que fueron dos conexiones? Vamos a dar una mirada más a fondo acá. Vamos a hacer lo siguiente. En cada método acaba de `cargarReporte` y de `listar`, nosotros vamos a imprimir la query que estamos ejecutando y ahí vamos a ver cuántas veces son ejecutadas estas queries.

[10:04] Vamos a empezar aquí por `cargaReporte`, voy a entrar acá, bueno, cómo llevamos el método `listar` vamos a categoría DAO. Acá en el `listar` voy a extraer esta query para una variable, entonces la hago acá. A ver, no lo tengo bueno, lo traigo acá. `Listar` variación `querySelect` = esto de acá, y ahí lo agrego acá, pero también voy a hacer un `System.out.println` de esta query para que la tengamos en la consola.

[10:45] Lo mismo. Voy a volver acá a `reporteFram` y lo mismo voy a hacer con el método `listar` de categoría. Entonces vamos a entrar acá en `productoDAO` `listar` por la categoría y vamos a hacer lo mismo. Voy a extraer esto acá. A ver, voy a extraer otra vez `querySelect`. Voy a poner el mismo nombre, como tenemos métodos distintos, no hay problema.

[11:11] querySelect, igual que nuestra query y voy a hacer un `System.out.println` también de esta query para ver en una consola cuántas veces estamos ejecutando este método. Vamos a reiniciar la aplicación y vamos a ver qué pasa. Ahí se levantó la aplicación, voy a limpiar la consola, vuelvo acá a la aplicación y ahora vamos a ejecutar reporte para ver cuántas veces se ejecutan las cookies.

[11:40] Acá hice un clic y se abrió el reporte y tenemos acá un `SELECT ID, NOMBRE FROM CATEGORÍA`. Una vez en la categoría, y una, dos, tres, cuatro selects a la tabla de producto. ¿Por qué? Como tenemos cuatro categorías acá en el reporte, nosotros fuimos cuatro veces a la base de datos para buscar a sus productos.

[12:03] Entonces acá nosotros estuvimos ejecutando cinco veces. O sea, tuvimos cinco conexiones a la base de datos para poder imprimir este reporte. Entonces para hacer una búsqueda entre dos tablas que son producto y categoría, estuvimos yendo varias veces a la base de datos, ahora en nuestro caso fueron pocas veces.

[12:23] Pero imaginemos un volumen mayor de datos o de relaciones. Ese número crece en n veces, es infinita la cantidad de veces. Y esto impacta directamente en la performance de nuestra aplicación, porque para cada iteración que estamos haciendo estamos abriendo una nueva conexión con la base de datos que tiene su costo.

[12:42] Esta solución que desarrollamos no está siguiendo las buenas prácticas de programación porque resolvimos un problema, pero creamos otros. Esta situación es conocida como queries $N + 1$, que es cuando para ejecutar una cierta funcionalidad, estamos yendo a la base de datos más de lo que es necesario cuando hay la posibilidad de ir una sola vez.

[13:04] Aquí estamos yendo a la base de datos en el método listar de categoría DAO, vamos acá en el `reporteFrame`, voy a agrandar un poquito más, estamos

yendo a la base de datos cuando llamamos el `cargaReporte` que llama el `listar` de `categoríaDAO` y también estamos yendo a la base de datos, cuando llamamos al método `listar` por categoría de producto DAO.

[13:27] O sea para cada categoría que estamos haciendo acá en el for estamos yendo a la base de datos de productos para hacer esta búsqueda. Entonces por eso que estamos haciendo cinco conexiones a la base de datos en nuestro escenario. ¿Hay una forma de arreglar esta situación? Aprovechando la relación que hay entre las entidades, categoría y producto.

[13:48] Por ahora quedamos por aquí para que piensen un poquito cómo debe ser esta solución. Pero si tienen un poco de apuro, vamos para la próxima clase para conocer esta solución.